Basic Text Analysis

Part-of-Speech Tagging

Parts of Speech

- 8–10 traditional parts of speech
  - Noun, verb, adjective, adverb, preposition, article, interjection, pronoun, conjunction, …
- Variously called:
  - Parts of speech, lexical categories, word classes, morphological classes, lexical tags, ...
- Lots of debate within linguistics about the number, nature, and universality of these
  - We'll completely ignore this debate

POS Examples

N	noun	*chair, bandwidth, pacing*
V	verb	*study, debate, munch*
ADJ	adjective	*purple, tall, ridiculous*
ADV	adverb	*unfortunately, slowly*
P	preposition	*of, by, to*
DET	determiner	*the, a, that, those*
INT	interjection	*ouch, hey*
PRO	pronoun	*I, me, mine*
CONJ	conjunction	*and, but, for, because*

POS Tagging

- The process of assigning a part-of-speech tag to every word of a sentence/text

WORD	TAG
the	DET
koala	N
put	V
the	DET
keys	N
on	P
the	DET
table	N

Why is POS Tagging Useful?

- First step of a vast number of practical tasks
- Speech synthesis
  - How to pronounce “lead”?
  - INsult inSULT
  - OBject obJECT
  - OVERflow overFLOW
  - DIScount disCOUNT
  - CONtent conTENT
- Parsing
  - Need to know if a word is an N or V before you can parse
- Information extraction
  - Finding names, relations, etc.
- Machine translation

Why is PoS Tagging Hard?

- Lexical ambiguity:
  - Secretariat/NNP is/VBZ expected/VBN to/TO **race**/VB tomorrow/NR
  - People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT **race**/NN for/IN outer/JJ space/NN
- Unknown words:
  - The/DT rural/JJ Babbitt/**???** who/WP bloviates/**???** about/IN progress/NN and/CC growth/NN

How is it Done?

- Two sources of information
  - Lexical information (the word itself)
    - Known words can be looked up in a lexicon listing possible tags for each word
    - Unknown words can be analyzed with respect to affixes, capitalization, special symbols, etc.
  - Contextual information (surrounding words)
    - A language model can rank tags in context
- Two main approaches
  - Rule-based systems
  - Statistical systems

POS Tagging
Choosing a Tagset

- There are so many potential distinctions we can draw
- To do POS tagging, we need to choose a standard set of tags to work with
- Could pick very coarse tagsets
  - N, V, Adj, Adv, …
- More commonly used sets are finer grained
  - English: Penn Treebank tagset, 45 tags
  - Swedish: SUC tagset, 25 base tags + features ≈ 150 tags
- Even more fine-grained tagsets exist

Open and Closed Classes

- Closed class: a small fixed membership
  - Prepositions: of, in, by, …
  - Pronouns: I, you, she, mine, his, this, that, …
  - Determiners: the, a, this, that, …
  - Usually function words
  - Often frequent and ambiguous
- Open class: new ones can be created all the time
  - English has 4: Nouns, Verbs, Adjectives, Adverbs
  - Many languages have these 4, but not all!
  - Usually content words
  - Often rare and (therefore sometimes) unknown

Penn TreeBank POS Tagset

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

How Hard is POS Tagging? Measuring Ambiguity

		87-tag Original Brown		45-tag Treebank Brown	
Unambiguous (1 tag)		**44,019**		**38,857**	
Ambiguous (2–7 tags)		**5,490**		**8844**	
Details:	2 tags	4,967		6,731	
	3 tags	411		1621	
	4 tags	91		357	
	5 tags	17		90	
	6 tags	2	(*well*, *beat*)	32	
	7 tags	2	(*still*, *down*)	6	(*well*, *set*, *round*, *open*, *fit*, *down*)
	8 tags			4	(*'s*, *half*, *back*, *a*)
	9 tags			3	(*that*, *more*, *in*)

The SUC POS Tagset

AB	Adverb	*inte*
DT	Determinerare	*denna*
HA	Frågande/relativt adverb	*när*
HD	Frågande/relativ determinerare	*vilken*
HP	Frågande/relativt pronomen	*som*
HS	Frågande/relativt possessivt pronomen	*vars*
IE	Infinitivmärke	*att*
IN	Interjektion	*ja*
JJ	Adjektiv	*glad*
KN	Konjunktion	*och*
NN	Substantiv	*pudding*
PC	Particip	*utsänd*
PL	Partikel	*ut*
PM	Egennamn	*Mats*
PN	Pronomen	*hon*
PP	Preposition	*av*
PS	Possessivt pronomen	*hennes*
RG	Grundtal	*tre*
RO	Ordningstal	*tredje*
SN	Subjunktion	*att*
UO	Utländskt ord	*the*
VB	Verb	*kasta*

The SUC POS Tagset

- The SUC tagset combines base tags with morphosyntactic features:

arenor	NN	UTR PLU IND NOM
arrogant	JJ	POS UTR SI
vinner	VB	PRS AKT
på	PP	
också	AB	
,	MID	
!	MAD	
(	PAD	

Feature	Value	Legend	Parts-of-speech where feature applies
Gender	UTR	Uter (common)	DT, HD, HP, JJ, NN, PC, PN, PS, (RG, RO)
	NEU	Neuter	
	MAS	Masculine	
Number	SIN	Singular	DT, HD, HP, JJ, NN, PC, PN, PS, (RG, RO)
	PLU	Plural	
Definiteness	IND	Indefinite	DT, (HD, HP, HS), JJ, NN, PC, PN, (PS, RG, RO)
	DEF	Definite	
Case	NOM	Nominative	JJ, NN, PC, PM, (RG, RO)
	GEN	Genitive	
Tense	PRS	Present	VB
	PRT	Preterite	
	SUP	Supinum	
	INF	Infinite	
Voice	AKT	Active	
	SFO	S-form (passive or deponential)	
Mood	KON	Subjunctive (Sw. konjunktiv)	
Participle form	PRS	Present	PC
	PRF	Perfect	
Degree	POS	Positive	(AB), JJ
	KOM	Comparative	
	SUV	Superlative	
Pronoun form	SUB	Subject form	PN
	OBJ	Object form	
	SMS	Compound (Sw. sammansättningsform)	All parts-of-speech

Evaluation

- So once you have you POS tagger running how do you evaluate it?
  - Overall error rate with respect to a manually annotated gold-standard test set
  - Error rates on known vs. unknown words
  - Error rates on particular tags
- Accuracy typically reaches 96–97% for English newswire text
  - What about Turkish?
  - What about twitter?

Error Analysis

- Look at a confusion matrix

	IN	JJ	NN	NNP	RB	VBD	VBN
IN	—	.2			.7		
JJ	.2	—	**3.3**	2.1	1.7	.2	**2.7**
NN		**8.7**	—				.2
NNP	.2	**3.3**	**4.1**	—	.2		
RB	**2.2**	2.0	.5		—		
VBD		.3	.5			—	**4.4**
VBN		**2.8**				**2.6**	—

- See what errors are causing problems
  - Noun (NN) vs ProperNoun (NNP) vs Adj (JJ)
  - Preterite (VBD) vs Participle (VBN) vs Adjective (JJ)

Two Methods for POS Tagging

1. Rule-based systems
   - Constraint Grammar
   - Transformation-Based Learning
2. Statistical sequence models
   - Hidden Markov Models
   - Maximum Entropy Markov Models
   - Conditional Random Fields

Rule-Based Tagging

- Constraint Grammar
  - Assign all possible tags to each word
  - Apply rules that discard tags based on context
  - Rules created by hand
- Transformation-Based Learning
  - Assign most frequent tag to each word
  - Apply rules that replace tags based on context
  - Later rules may overwrite earlier rules
  - Rules learned from tagged corpus

Constraint Grammar

Pavlov	N NOM SG PROPER
had	V PAST VFIN
	PCP2

Rule 2: Discard ADV before N

Rule 1: Discard PCP2 before PCP2

	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	N NOM SG

...

Constraint Grammar

- Advantages:
  - Can achieve very high recall with good lexical resources
  - Rules can be interpreted by humans, which facilitates debugging
- Drawbacks:
  - Not always possible to eliminate all ambiguity
  - Rules design is difficult and time-consuming

TBL Tagging

- Initial-state annotation:
  - Known words: most likely tag
  - Unknown words: NNP if capitalized, else NN
- Apply transformation rules in sequence:
  - Replace tag A with tag B in context C

TBL Learning

- Given a tagged training corpus:
  - Apply initial-state annotation
  - Repeat until no improvement:
    - Consider all possible transformation rules
    - Select rule with best score on training data
    - Add rule to the end of rule sequence
- Design decisions:
  - Rule templates
    - For example: n-word window of words/tags
  - Scoring function
    - For example: error reduction on training set

Top Rules Learnt for English

From	To	If
NN	VB	previous tag is TO
to/TO conflict/NN→ NB		
VBP	VB	one of the previous 3 tags is MD
might/MD vanish/VBP→ VB		
NN	VB	one of the previous two tags is MD
might/MD not reply/NN→ VB		
VB	NN	one of the previous two tags is DT
the/DT amazing play/VB→ NN		

TBL Evaluation

- Advantages:
  - Rules can be interpreted by humans, which facilitates debugging
  - Rules are learnt automatically from data
- Drawbacks:
  - Not quite as accurate as the best models
  - Slow to train on large data sets

Hidden Markov Model Tagging

- Using an HMM to do POS tagging is a special case of *Bayesian inference*
  - Given a sequence of tags:

    Secretariat is expected to race tomorrow
  - What is the best sequence of tags?
- Probabilistic view:
  - Consider all possible sequences of tags
  - Choose the most probable one given $w_1 \ldots w_n$

Getting to HMMs

- We want, out of all sequences of n tags $t_1 \ldots t_n$ the single tag sequence such that $P(t_1 \ldots t_n | w_1 \ldots w_n)$ is highest.

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$$

- Hat ^ means "our estimate of the best one"
- Argmax_x f(x) means "the x such that f(x) is maximized"

Getting to HMMs

- This equation is guaranteed to give us the best tag sequence

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$$

- But how to make it operational? How to compute this value?
- Strategy in Bayesian classification:
  - Use Bayes rule to break down the problem
  - Make appropriate independence assumptions

Using Bayes Rule

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

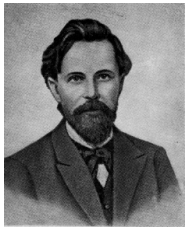
$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \frac{P(w_1^n|t_1^n)P(t_1^n)}{P(w_1^n)}$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(w_1^n|t_1^n)P(t_1^n)$$

Likelihood and Prior

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \overbrace{P(w_1^n|t_1^n)}^{\text{likelihood}} \overbrace{P(t_1^n)}^{\text{prior}}$$

$$P(w_1^n|t_1^n) \approx \prod_{i=1}^{n} P(w_i|t_i)$$



$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1})$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i) P(t_i|t_{i-1})$$

Two Kinds of Probabilities

- Tag transition probabilities $p(t_i|t_{i-1})$
  - Dets likely to precede adjs and nouns
    - That/DT flight/NN
    - The/DT yellow/JJ hat/NN
    - So we expect P(NN|DT) and P(JJ|DT) to be high
    - But P(DT|JJ) to be low
  - We can estimate P(NN|DT) from tagged corpus:

$$P(NN|DT) = \frac{C(DT,NN)}{C(DT)} = \frac{56,509}{116,454} = .49$$

- This is a bigram language model over tags!

Two Kinds of Probabilities

- Word likelihood probabilities $p(w_i|t_i)$
  - VBZ (3sg Pres verb) likely to be "is"
  - Estimate P(is|VBZ) from tagged corpus:

$$P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = .47$$

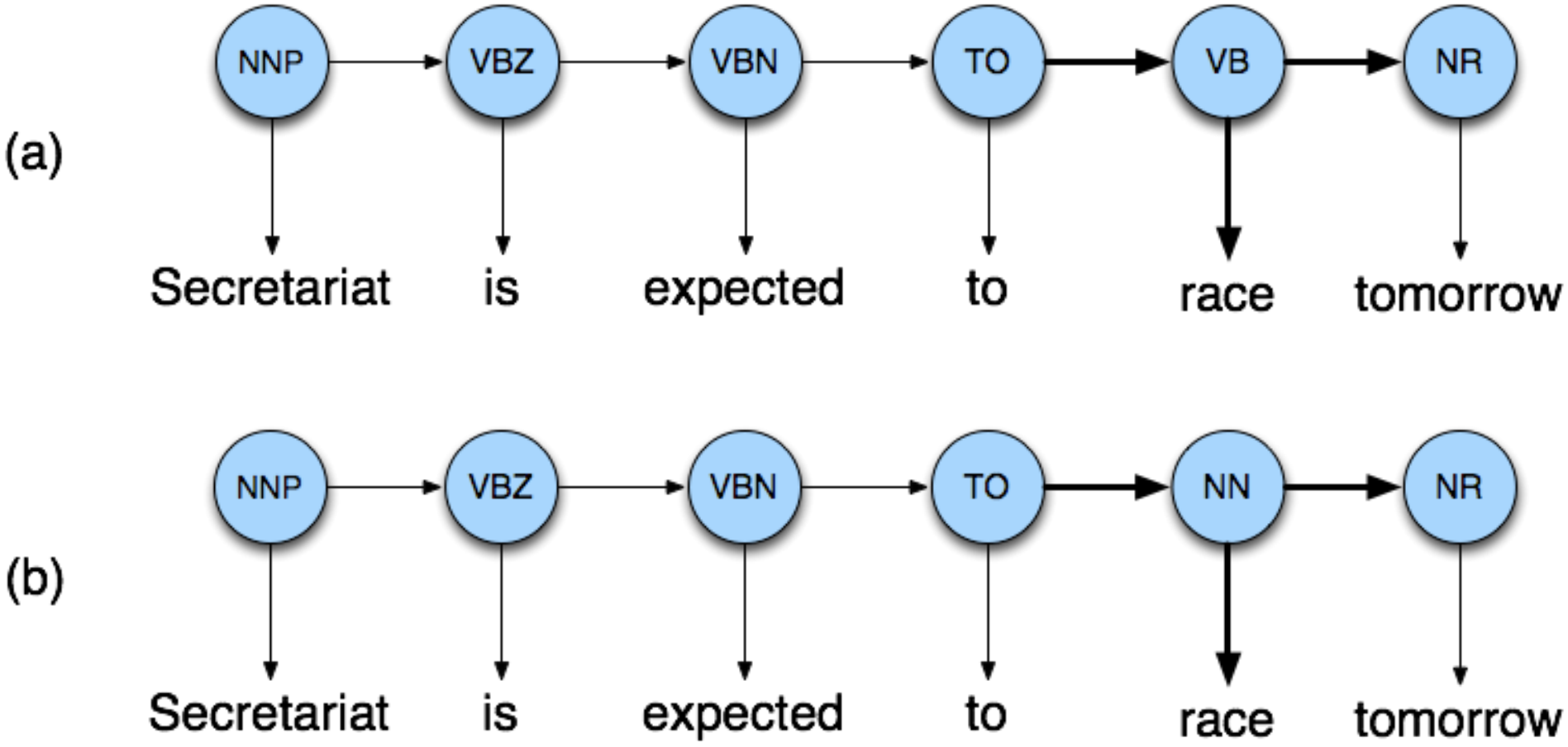
Example: The Verb "race"

Secretariat/NNP is/VBZ expected/VBN to/TO **race**/VB tomorrow/NR

People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT **race**/NN for/IN outer/JJ space/NN

How do we pick the right tag?

Disambiguating “race”



Example

P(NN|TO) = .00047
P(race|NN) = .00057
P(NR|NN) = .0012

P(VB|TO) = .83
P(race|VB) = .00012
P(NR|VB) = .0027

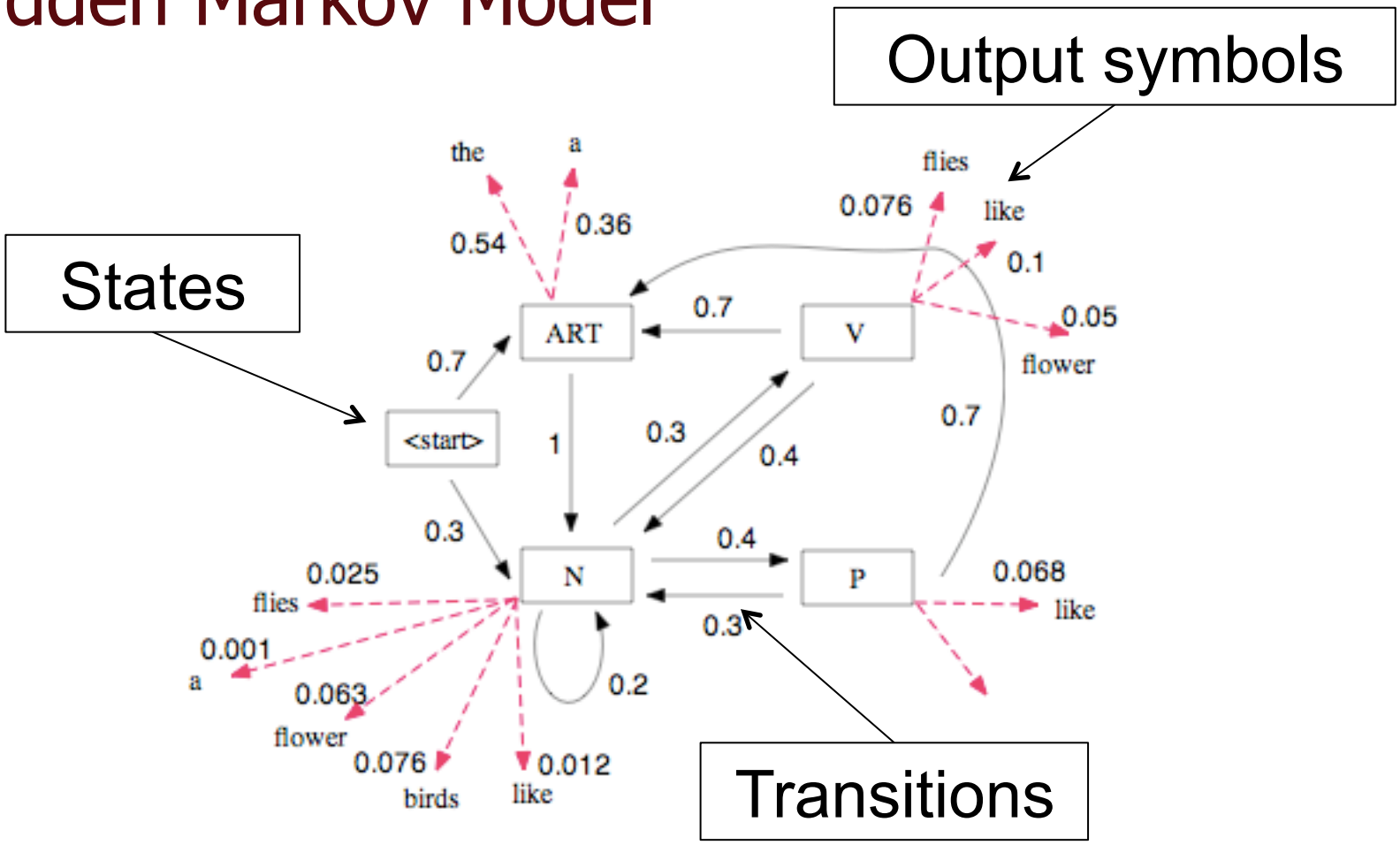
P(NN|TO) x P(NR|NN) x P(race|NN)=.00000000032

P(VB|TO) x P(NR|VB) x P(race|VB) = .00000027

So we (correctly) choose the verb reading!

Hidden Markov Models

- What we have described is a (first-order) Hidden Markov Model



More Formally

- Alphabet $\Sigma = \{ s_1, s_2, \ldots, s_M \}$
- Set of states $Q = \{ q_1, q_2, \ldots, q_M \}$
- Transition probabilities between any two states
  $a_{ij} = P(q_j \mid q_i)$ = transition prob from state i to state j
- Start probabilities for any state
  $\pi_{0i} = P(q_i)$ = start prob for state I
- Emission probabilities for each symbol and state
  $b_{ik} = P(s_k \mid q_i)$

Looks Familiar?

- HMMs are like FSAs except:
  - Transitions and emissions are decoupled
    - The model first transitions to a state, then emits a symbol in that state
  - Transitions and emissions are probabilistic

Problems for HMM

- Learning:
  - How to estimate transition and emission probabilities
- Inference/Decoding:
  - How to find most probable state sequence for a given observation sequence
- Today:
  - Learning from a tagged corpus
- Next time:
  - Decoding + learning from raw text

Supervised Learning

- Supervised learning:
  - Learning from a tagged corpus
- Start probabilities:
  - Eliminate by introducing dummy state <start> with P(<start>) = 1
- Transition probabilities:
  - N-gram language model over tags
  - States represent context
    - First-order (bigram) model, state = tag unigram
    - Second-order (trigram) model, state = tag bigram

Transition Probabilities

- Compute tag n-gram counts from corpus
  - Bigram case:

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

- Use smoothing to handle unseen n-grams
  - Distribution less sparse than for word n-grams
  - Laplace smoothing works okay
  - Backoff and interpolation are common

Emission Probabilities

- Compute word-tag counts from corpus:

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

- Complications:
  - Known words with unknown tags?
  - Unknown words
    - Suffix probabilities work well for many languages
    - Additional features: capitalization, numbers, etc.

Summary

- Part-of-speech tagging
  - Basic step in many analysis pipelines
  - Different tagsets for different languages and applications
- Methods
  - Rule-based systems (CG, TBL)
  - Statistical sequence models (HMM, …)
- State of the art:
  - 96-97% accuracy for English newswire text